Journal of Biomedical Informatics 58 (2015) 122–132

Contents lists available at ScienceDirect

Journal of Biomedical Informatics

journal homepage: www.elsevier.com/locate/yjbin

Text mining for pharmacovigilance: Using machine learning for drug name recognition and drug–drug interaction extraction and classification

Asma Ben Abacha [a,*], Md. Faisal Mahbub Chowdhury [b], Aikaterini Karanasiou [a], Yassine Mrabet [a], Alberto Lavelli [c], Pierre Zweigenbaum [d]

[a] Luxembourg Institute of Science and Technology, Luxembourg
[b] IBM Research, NY, USA
[c] HLT Research Unit, FBK, Trento, Italy
[d] LIMSI-CNRS, Orsay, France

ARTICLE INFO

Article history:
Received 30 October 2014
Revised 31 August 2015
Accepted 22 September 2015
Available online 30 September 2015

Keywords:
Text mining
Machine learning
Drug name recognition
Drug–drug interactions
Pharmacovigilance

ABSTRACT

Pharmacovigilance (PV) is defined by the World Health Organization as the science and activities related to the detection, assessment, understanding and prevention of adverse effects or any other drug-related problem. An essential aspect in PV is to acquire knowledge about Drug–Drug Interactions (DDIs). The shared tasks on DDI-Extraction organized in 2011 and 2013 have pointed out the importance of this issue and provided benchmarks for: Drug Name Recognition, DDI extraction and DDI classification. In this paper, we present our text mining systems for these tasks and evaluate their results on the DDI-Extraction benchmarks. Our systems rely on machine learning techniques using both feature-based and kernel-based methods. The obtained results for drug name recognition are encouraging. For DDI-Extraction, our hybrid system combining a feature-based method and a kernel-based method was ranked second in the DDI-Extraction-2011 challenge, and our two-step system for DDI detection and classification was ranked first in the DDI-Extraction-2013 task at SemEval. We discuss our methods and results and give pointers to future work.

1. Introduction

Drug–Drug Interaction (DDI) is a condition when one drug influences the level or activity of another drug. Acquiring knowledge of DDIs has significant importance for both patient safety and efficient health care management. It was reported that about 2.2 million people in USA, age 57–85, were taking potentially dangerous combinations of drugs [23]. It was also estimated that deaths from accidental drug interactions rose by 68% in a 5-years period between 1999 and 2004 [30]. The interactions between drugs can also increase the risk of side effects, and their number was shown to be correlated to the number of drugs taken by each patient [20]. However, with the rapid growth of drug industries and the exponential number of possible combinations, keeping strong insights on DDIs is becoming a more and more challenging task.

Biomedical literature and clinical reports provide a natural ground to detect and analyze DDIs at a big scale. However making use of such huge quantity of information requires the design of efficient and automatic tools that can assist human experts in the discovery and follow-up of DDIs. The DDI-Extraction-2011 and DDI-Extraction-2013 shared tasks particularly underlined the importance of the extraction of DDIs from medical texts.

Recognizing DDIs from medical texts requires (i) identifying drug mentions, (ii) detecting the expressions that indicate interactions between the mentioned drugs, and (iii) classifying of the interactions according to their types. In this paper we present three approaches to tackle these tasks:

- A feature-based approach for drug name recognition, evaluated on the DDI-2013 corpus for Task 9.1: Drug name recognition.
- A hybrid approach for DDI detection [6], combining a feature-based method and a kernel-based method, evaluated on the DDI-2011 corpus.

* Corresponding author.
E-mail addresses: asma.benabacha@list.lu (A. Ben Abacha), mchowdh@us.ibm.com (Md. Faisal Mahbub Chowdhury), aikaterini.karanasiou@list.lu (A. Karanasiou), yassine.mrabet@list.lu (Y. Mrabet), lavelli@fbk.eu (A. Lavelli), pz@limsi.fr (P. Zweigenbaum).

http://dx.doi.org/10.1016/j.jbi.2015.09.015
1532-0464/

- A two-step approach for DDI detection and classification [12] which is basically a considerable improvement on the kernel method mentioned above. This approach was evaluated both on the DDI-2013 corpus for Task 9.2: Extraction of drug–drug interactions and on the DDI-2011 corpus.

In the following we first survey research works related to the above three tasks. We then present our approaches in Sections 3–5. The results on the data sets of the DDI-Extraction-2011 and DDI-Extraction-2013 challenges are described in Sections 6 and 7 respectively. We finally conclude with pointers to future work in Section 8.

2. Related works

Several approaches have been proposed for the recognition of medical entities such as diseases, treatments and exams [2]. Other efforts tackled the extraction of specific entities such as drugs. For instance, the 2009 i2b2 challenge[1] focused on the extraction of medication-related information (e.g. medication name, dosage, frequency, mode of administration) from narrative patient records.

Several approaches have also studied the recognition of chemical entities (i.e. chemical compounds and drugs). Grego et al. [17] proposed a chemical entity recognition approach using Conditional Random Fields for identifying chemical terms and lexical similarity for the classification of entities according to the ChEBI ontology. They participated to the SemEval-2013 challenge on the recognition and classification of drug names (Task 9.1) and obtained a macro-average F_1 score of 0.577 on the full dataset (DrugBank and Medline).

Rocktäschel et al. [31] studied the impact of domain-specific features on the task of recognizing and classifying mentions of pharmacological substances. They used predictions of their improved version of the ChemSpot tool[2] [32] and features derived from (i) Jochem,[3] a dictionary for the identification of small molecules and drugs in text [18], (ii) the PHARE ontology [14] and (iii) the ChEBI ontology [25]. Their system was ranked first in the SemEval-2013 Task 9.1 with a macro-average F_1 score of 0.652 on the whole dataset (MedLine and DrugBank).

Besides the recognition of textual mentions of medical entities, the detection and classification of (bio) medical relations is an important task that was addressed by many research works [1].

For instance, Song et al. [37] proposed a protein–protein interaction (PPI) extraction technique called PPISpotter that combines an active learning technique with semi-supervised Support Vector Machines (SVM) to extract protein–protein interaction. Chen et al. [5] proposed a PPI Pair Extractor (PPIEor), based on a SVM for binary classification which uses a linear kernel and a rich set of features based on linguistic analysis, contextual words, interaction words, interaction patterns and specific domain information. Li et al. [24] use an ensemble kernel to extract the PPI information. This ensemble kernel consists of a feature-based kernel and a structure-based kernel using the parse trees of the sentences containing at least two protein names.

Other approaches particularly focused on the extraction of DDI. Segura-Bedmar et al. [34] compared two different approaches for the extraction of DDIs from texts: (i) a hybrid linguistic approach that combines shallow parsing and pattern matching and (ii) a kernel-based approach that uses SVM presented by Giuliano et al. [16]. For the evaluation, they created and annotated the first corpus annotated with DDIs containing 579 documents from the DrugBank database and a total of 3160 DDIs. The lexical patterns achieve 67.30% precision and 14.07% recall. With the inclusion of appositions and coordinate structures they obtained 48.69% precision and 25.70% recall. The second approach based on kernel-methods achieves better performance with 55.1% precision and 82.3% recall.

Also, different machine learning approaches were proposed within the DDI-Extraction challenges in 2011 and 2013. In [36], the authors observed that in the 2013 DDI-Extraction task non-linear kernel-based methods outperformed linear SVM-based approaches.

In this paper, we describe our experience from the participation in the 2011 and 2013 DDI-Extraction tasks, and analyze the results obtained by (i) our hybrid system for DDI detection [6] which combines feature-based and kernel-based methods (Sections 4 and 6) and (ii) our two-step approach for the detection and classification of DDIs [12] (Sections 5 and 7.3). We also present our new approach for drug name recognition in Section 3 and evaluate it on the DDI-Extraction-2013 corpus in Section 7.2.

3. Feature-based method for drug name recognition

Drug recognition from medical texts involves two main tasks: (i) identification of mentions boundaries in the sentences and (ii) entity classification. In the context of the DDI-Extraction-2013 challenge, four classes were considered for the task of the recognition and classification of drug names: Drug, Drug_n, Brand and Group (cf. Section 7.2 for the description of each category). We proposed and evaluated a feature-based method using the Conditional Random Fields (CRF) algorithm and several linguistic and semantic features. In the following section we present the main characteristics of the CRF algorithm and its application in the scope of the challenge. In Sections 3.2 and 3.3 we present the final set of linguistic and semantic features that we selected to train the CRF-based classifier.

3.1. CRF algorithm

Words in a sentence form a sequence and the decision on a word's category can be influenced by the decision on the category of the preceding word. This dependency is taken into account in sequential models such as Hidden Markov Models (HMM) or Conditional Random Fields (CRF). In contrast with HMM, CRF learning maximizes the conditional probability of classes with respect to observations rather than their joint probability. This makes it possible to use any number of features which may be related to all aspects of the input sequence of words. These properties are assets of CRF for several NLP tasks, such as POS tagging, noun phrase chunking, or named entity recognition.

We use the CRF learning algorithm [22] in order to annotate the words with BIO (Beginning-Inside-Outside) labels. Given n entity types (e.g. Drug, Drug_n, Brand, Group), we consider n classes of type 'B' and n classes of type 'I'. The B_Entity_Type represents the first word of a pharmacological entity, I_Entity_Type represents the remaining words of the pharmacological entity and the O represents the words that are not terms of a pharmacological entity. For instance, the words of the following sentence will be annotated as:

"Studies(O) have(O) shown(O) that(O) TIKOSYN(B-Drug) does(O) not(O) affect(O) the(O) pharmacokinetics(O) of(O) oral(B-Drug) contraceptives(I-Drug)".

According to [26], suppose $x = \{x_1, x_2, x_3, \ldots, x_T\}$ is a set of input values (*e.g.* a sequence of words) and $s = \{s_1, s_2, s_3, \ldots, s_T\}$ is a set of states that are assigned to named entity labels, CRF estimates the conditional probability of a state sequence given an input sequence as follows:

$$P(s|x) = \frac{1}{Z} exp\left(\sum_{t=1}^{T}\sum_{k=1}^{K}\lambda_k f_k(s_{t-1}, s_t, x, t)\right)$$

[1] https://www.i2b2.org/NLP/Medication/.
[2] https://www.informatik.hu-berlin.de/forschung/gebiete/wbi/resources/chemspot/chemspot/.
[3] http://biosemantics.org/index.php?page=Jochem.

indicating whether the current word and where $1,\ldots,T$ represent the word positions, $1,\ldots,K$ represent the positions of the weighted features, the f_k represents the feature function and the λ_k is the weight of each feature function.

We use the CRF++[4] tool for constructing probabilistic models over the training data in order to predict the entity type of tokens in the test data. We use the values of several features as an input set of observations (e.g. words, lemmas, POS-tag, length of words). The final list of features is presented in the following sections.

3.2. Token and linguistic features

Token features:

- The original form and the lemma of the current word.
- Two tokens after and two tokens before the current word and their lemmas.

Linguistic features:

- The part-of-speech tag (POS-tag) of the current word, its preceding two words and its following two words.
- The length of the current word, of the previous word and of the two following words.
- The suffix of the current word (3 last characters).

Binary features indicating whether:

- The current word is a number.
- The current word and the previous word are composed by alphabetic characters only.
- The current word, the previous word and the next word begin with a capital letter.
- The current word and the two previous words are all capitalized and without digits.
- The current word, the two previous words and the next word have length equal to 2.
- The current word has lower letters mixed with capital letters and does not contain digits.
- The current word and the next word contain a forward slash.

3.3. Semantic features

- Several binary features are defined to indicate the presence/absence of a word in a specific list. The considered lists are: a stop words list,[5] a list of abbreviations[6] and a list of medical units of measurement.[7]
- A list of drugs, taken from drugs@FDA[8] is used to tag the current word, the two previous words and the next word as a **Drug**. More precisely, the BIO format is used to annotate the words: B-Drug for the first word of a drug name, I-Drug for the words inside a drug name and O for the words that are not in the list of drugs.
- A list of drugs' ingredients, taken from RxTerms-drug interface terminology,[9] is used to annotate the current word as an **Ingredient**. Similarly to the previous annotation method, the words of medical texts are annotated using the BIO format: B-Ingredient for the first word of an ingredient name, I-Ingredient for the words inside an ingredient name and O for the words that are not contained in the list of ingredients.

The evaluation of our Drug Name Recognition approach is presented in Section 7.2. In the following section we present our first approach for drug–drug interaction extraction.

4. Hybrid approach for drug–drug interactions extraction (FBM–KBM)

Our hybrid approach for DDI extraction combines: (i) a feature-based machine learning method and (ii) a kernel based method. We tested both the union and the intersection of the results of each method.

4.1. Feature-based machine learning method (FBM)

In this approach, the problem is modeled as a supervised binary classification task. We used a SVM classifier to decide whether a candidate DDI pair is an authentic DDI or not. We used the LibSVM tool [3] to train a model using C-Support Vector Classification (C-SVC) with the Radial Basis kernel function. The particular SVM implementation (i.e. C-SVC SVM) and values of the associated parameters are selected by doing experiments on a small subset of the data. The set of features we used are described in Sections 4.1.1 and 4.1.2.

4.1.1. Features for DDI extraction

The following set of features was used to describe each candidate DDI pair (D1,D2):

- **Word Features.** Include words of D1, words of D2, words between D1 and D2 and their number, three words before D1, three words after D2 and lemmas of all these words.
- **Morphosyntactic Features.** Include Part-of-speech (POS) tags of each drug word (D1 and D2), POS of the previous three and next three words. We use TreeTagger [10] to obtain lemmas and POS tags.
- **Other Features.** Include, among others, verbs between D1 and D2 and their number, first verb before D1 and first verb after D2.

4.1.2. Advanced features

In order to improve the performance of our system, we used lists of interacting drugs, constructed by extracting drug pairs that are related by an interaction in the training corpus. We defined a feature to represent the fact that candidate drug pairs are present in this list.

However, such lists are not sufficient to identify an interaction between new drug pairs. We also worked on detecting keywords expressing such relations in the training sentences. The following examples of positive (1,2) and negative (3) sentences show some of the keywords or trigger words that may indicate an interaction relationship.

1. *The oral bioavailability of enoxacin is* **reduced** *by 60% with* **coadministration** *of ranitidine.*
2. *Etonogestrel may* **interact** *with the following medications: acetaminophen (Tylenol) . . .*
3. *There have been* **no** *formal studies of the* **interaction** *of Levulan Kerastick for Topical Solution with any other drugs . . .*

[4] http://crfpp.googlecode.com/svn/trunk/doc/index.html.
[5] http://www.ranks.nl/stopwords.
[6] http://en.wikipedia.org/wiki/List_of_medical_abbreviations.
[7] http://www.mayomedicallaboratories.com/test-catalog/appendix/measurement.html.
[8] http://www.accessdata.fda.gov/scripts/cder/drugsatfda.
[9] https://wwwcf.nlm.nih.gov/umlslicense/rxtermApp/rxTermFileStructure.cfm.
[10] http://www.cis.uni-muenchen.de/~schmid/tools/TreeTagger/.

To exploit this "semantic" information, we defined the following features:

- **Trigger words.** This category of features indicates whether a specific trigger word (e.g. induce, inhibit) occurs in the sentence. The trigger words were collected manually from the training corpus.
- **Negation.** This category of features indicates if a negation (e.g. not, no) is detected at a limited distance of characters before, between and after the two considered drugs.

4.2. Kernel-based machine learning method (KBM)

In this approach, the DDI extraction task was addressed using a system that exploits kernel-based method. Initially, the data had been pre-processed to obtain relevant information on the tokens in the sentences.

4.2.1. Data pre-processing

We used the Stanford parser[11] [21] for tokenization, POS-tagging and parsing of the sentences. The SPECIALIST lexicon tool[12] was used to normalize tokens to avoid spelling variations and also to provide lemmas. The dependency relations produced by the parser were used to create dependency parse trees for the corresponding sentences.

4.2.2. System description

Our kernel-based system uses a composite kernel K_{SMP} which combines multiple tree and feature-based kernels. It is defined as follows:

$$K_{SMP}(R_1,R_2) = K_{SL}(R_1,R_2) + w_1^* K_{MEDT}(R_1,R_2) + w_2^* K_{PST}(R_1,R_2)$$

where K_{SL}, K_{MEDT} and K_{PST} represent respectively shallow linguistic (SL) [16], mildly extended dependency tree (MEDT) [13] and PST [28] kernels, and w_i represents multiplicative constant(s). The values for all of the w_i used during our experiments were equal to 1.[13] The composite kernel is valid according to the kernel closure properties.

A dependency tree (DT) kernel, pioneered by Culotta and Sorensen [15], is typically applied to the minimal or smallest common subtree of a dependency parse tree that includes a target pair of entities. Such subtree reduces unnecessary information by placing word(s) closer to its dependent(s) inside the tree and emphasizes local features of the corresponding relation. However, sometimes a minimal subtree might not contain important cue words or predicates. The MEDT kernel addresses this issue using some linguistically motivated expansions. We used the best settings for the MEDT kernel reported by Chowdhury et al. [13] for protein–protein interaction extraction.

The PST kernel is basically the path-enclosed tree (PET) proposed by Moschitti [28]. This tree kernel is based on the smallest common subtree of a phrase structure parse tree, which includes the two entities involved in a relation.

The SL kernel is perhaps the best feature-based kernel used so far for biomedical RE tasks (e.g. PPI and DDI extraction). It is a combination of global context (GC) and local context (LC) kernels. The GC kernel exploits contextual information of the words occurring before, between and after the pair of entities (to be investigated for RE) in the corresponding sentence; while the LC kernel exploits contextual information surrounding individual entities.

The jSRE system[14] is the implementation of these kernels using the support vector machine (SVM) algorithm. It should be noted that, by default, the jSRE system uses the ratio of negative and positive examples as the value of the cost-ratio-factor[15] parameter during SVM training.

Segura-Bedmar et al. [35] used the jSRE system for DDI extraction on the same corpus (in the MMTx format) that has been used during the DDI-Extraction-2011 challenge. They experimented with various parameter settings, and reported as much as an F_1 score of 0.6001. We used the same parameter settings (*n*-gram = 3, window-size = 3) with which they obtained their best result.

To compute the feature vectors of SL kernel, we used the jSRE system. The tree kernels and composite kernel were computed using the SVM-LIGHT-TK toolkit[16] [29,19]. Finally, the ratio of negative and positive examples has been used as the value of the cost-ratio-factor parameter.

5. Two-step approach for DDI detection and classification

Our second DDI extraction approach is basically a considerable improvement (in terms of results obtained for DDI extractions) on our hybrid approach described earlier. It performs DDI detection and classification in two separate steps. We first present the DDI detection method consisting in (i) discarding less informative sentences, (ii) discarding less informative instances, and (iii) training the system (a single model regardless of DDI types) on the remaining training instances and identifying possible DDIs from the remaining test instances.

5.1. Exploiting the scope of negations for sentence filtering

Negation is a linguistic phenomenon where a *negation cue* (e.g. *not*) can alter the meaning of a particular text segment or of a fact. This text segment (or fact) is said to be inside the *scope of such negation (cue)*. In one of our recent papers [11], we proposed an approach to exploit the scope of negations for RE. We hypothesize that a classifier trained solely on features related to the scope of negations can be used to pro-actively filter groups of instances which are less informative and mostly negative.

To be more precise, we propose to train a classifier (which will be applied before using the kernel based RE classifier mentioned in Section 5.3) that would check whether all the target entity mentions inside a sentence along with possible relation clues (or trigger words), if any, fall (directly or indirectly) under the scope of a negation cue. If such a sentence is found, then it would be identified as less informative and discarded (i.e. the candidate mention pairs inside such sentence would not be considered). During training (and testing), we group the instances by sentences. *Any sentence that contains at least one relation of interest is considered by the Less Informative Sentence (LIS) classifier as a positive (training/test) instance.* The remaining sentences are considered as negative instances.

We use a number of features related to negation scopes to train a binary SVM classifier that filters out less informative sentences Chowdhury and Lavelli [11]. These features are basically contextual and shallow linguistic features, and are described below:

- *has2TM:* The sentence has exactly 2 target entity mentions.

[11] http://nlp.stanford.edu/software/lex-parser.shtml.
[12] http://lexsrv3.nlm.nih.gov/SPECIALIST/index.html.
[13] Due to time constraints, we did not do tuning of the multiplicative constant(s) (i.e. w_i) during DDI-Extraction-2011. However, we did perform parameter tuning using 5-fold cross-fold validation on the training data for DDI-Extraction-2013 which will be explained later.
[14] http://hlt.fbk.eu/en/technology/jSRE.
[15] This parameter value is the one by which training errors on positive examples would outweigh errors on negative examples.
[16] http://disi.unitn.it/moschitti/Tree-Kernel.htm.

- *has3OrMoreTM:* The sentence has more than 2 target entity mentions.
- *allTMonRight:* All target entity mentions inside the sentence appear after the negation cue.
- *neitherAllTMonLeftOrRight:* Some but not all target entity mentions appear after the negation cue.
- *negCue:* The negation cue itself.
- *immediateGovernor:* The word on which the cue is directly syntactically dependent.
- *nearestVerbGovernor:* The nearest verb in the dependency graph on which the cue is syntactically dependent.
- *isVerbGovernorRoot:* The *nearestVerbGovernor* is root of the dependency graph of the sentence.
- *allTMdependentOnNVG:* All target entity mentions are syntactically dependent (directly/indirectly) on the *nearestVerbGovernor*.
- *allButOneTMdependentOnNVG:* All but one target entity mentions are syntactically dependent on the *nearestVerbGovernor*.
- *although∗PrecedeCue:* The syntactic clause containing the negation cue begins with *"although/ though/ despite/ in spite"*.
- *commaBeforeNextTM:* There is a comma in the text between the negation cue and the next target entity mention after the cue.
- *commaAfterPrevTM:* There is a comma in the text between the previous target entity mention before the negation cue and the cue itself.
- *sentHasBut:* The sentence contains the word *"but"*.

The objective of the classifier is to decide whether all target entity mentions as well as any possible evidence inside the corresponding sentence fall under the scope of a negation cue in such a way that the sentence is unlikely to contain the relation of interest (e.g. DDI). If the classifier finds such a sentence, it is assigned the negative class label.

At present, we focus only on the first occurrence of the negation cues "no", "n't" or "not". These cues usually occur more frequently and generally have larger negation scope than other negation cues.

The LIS classifier is trained using a linear SVM classifier. Its hyper-parameters[17] are tuned during training using 5-fold cross-validation for obtaining maximum recall with a non-zero precision. In this way we minimize the number of false negatives (i.e. sentences that contain relations but are wrongly filtered out). Once the classifier is trained using the training data, we apply it on both the training and test data. However, if the recall of the LIS classifier is found to be below a *threshold value*[18] during cross validation on the training data of a corpus, it is not used for sentence filtering on such corpus.

Any (training/test) sentence that is classified as negative is considered as a less informative sentence and is filtered out. In other words, such a sentence is not considered for RE. However, it should be noted that, if such a sentence is a test sentence and it contains positive RE instances, then *all these filtered positive RE instances are automatically considered as false negatives during the calculation of RE performance*.

We rule out sentences (i.e. we consider them neither positive nor negative instances for training the classifier that filters less informative sentences) during both training and testing if any of the following conditions holds:

- The sentence contains less than two target entity mentions (such sentence would not contain the relation of interest anyway).
- It has any of the following phrases – "not recommended", "should not be" or "must not be".[19]
- There is no "no", "n't" or "not" in the sentence.
- No target entity mention appears in the sentence after "no", "n't' or "not".

5.2. Discarding instances using semantic roles and contextual evidence

For identifying less informative negative instances, we exploit static (i.e. already known, heuristically motivated) and dynamic (i.e. automatically collected from the data) knowledge which has been proposed in [10]. This knowledge is described by the following criteria:

- **C1:** If each of the two entity mentions (of a candidate pair) has *anti-positive governors* with respect to the type of the relation, then they are not likely to be in a given relation.
- **C2:** If two entity mentions in a sentence refer to the same entity, then it is unlikely that they would have a relation between themselves.
- **C3:** If a mention is the abbreviation of another mention (i.e. they refer to the same entity), then they are unlikely to be in a relation.

Criteria C2 and C3 (static knowledge) are quite intuitive. For criterion C1, we construct on the fly a list of *anti-positive governors*, to be discussed below, taken from the training data and use them for detecting pairs that are unlikely to be in relation. As for criterion C2, we simply check whether two mentions have the same name and there is more than one character between them. For criterion C3, we look for any expression of the form "Entity1 (Entity2)" and consider "Entity2" as an abbreviation or alias of "Entity1".

The above criteria are used to filter instances from both training and test data. *Any positive test instance filtered out by these criteria is automatically considered as a false negative during the calculation of RE performance*.

Anti-positive governors: The semantic roles of the entity mentions may indirectly contribute either to relate or not to relate them in a particular relation type (e.g. PPI) in the corresponding context. To put it differently, the semantic roles of two mentions in the same context could provide an indication whether the relation of interest does *not* hold between them. Interestingly, the word on which a certain entity mention is (syntactically) dependent (along with the dependency type) could often provide a clue of the semantic role of such mention in the corresponding sentence. Our goal is to automatically identify the words (if any) that tend to prevent mentions, which are directly dependent on those words, from participating in a certain relation of interest with any other mention in the same sentence. We call such words ***anti-positive governors*** and assume that they could be exploited to identify negative instances (i.e. negative entity mention pairs) in advance. Below we describe our approach for the automatic identification of such words.

Let $\mathbb{EN}$ be the set of entity mentions such that if $e_s^i \in \mathbb{EN}$ (where s indicates the corresponding training sentence and i indicates the corresponding entity mention index inside such sentence), then e_s^i does not have any relation of interest (i.e. PPI) with any other mention inside the same sentence.

Let $\mathbb{EP}$ be the set of entity mentions such that if $e_s^k \in \mathbb{EP}$ (where s indicates the corresponding training sentence and k indicates the

[17] (i) C: trade-off between training error and margin, and (ii) cost: cost-factor, by which training errors on positive examples outweight errors on negative examples.

[18] We set it to 70.0 which is chosen empirically.

[19] These expressions often provide clues that one of the drug entity mentions negatively influences the level of activity of the other.

corresponding entity mention index inside such sentence), then e_s^k has at least one relation of interest with one of the mentions inside the same sentence.

For example, consider the following sentence (taken from the IEPA corpus) where there are three entity mention annotations – *oxytocin*[1], *oxytocin*[2] and *IP3*[3].

> *These results indicate that oTP-1 may prevent luteolysis by inhibiting development of endometrial responsiveness to* ***oxytocin***[1] *and, therefore, reduce* ***oxytocin***[2]*-induced synthesis of* ***IP3***[3] *and PGF2 alpha.*

Here, the mention *oxytocin*[1] does not participate in any PPI relation in this sentence. So, it would be included in EN. The other two mentions would be added to EP, because they are in PPI relation with each other. Note that the two mentions of the entity *oxytocin* are treated separately.

Now, let GV be the set of governor words where for each $w \in GV$, (*i*) there is at least one mention $e_s^i \in EN$ which is syntactically dependent on w in the corresponding training sentence *s* and (*ii*) there is *no* mention $e_s^k \in EP$ which is syntactically dependent on w in the corresponding training sentence *s*. We call this set GV as the list of *anti-positive governors* [10].

5.3. Hybrid kernel-based RE classifier

As RE classifier we use the following hybrid kernel that has been proposed in [11]. It is defined as follows:

$$K_{Hybrid}(R_1, R_2) = K_{HF}(R_1, R_2) + K_{SL}(R_1, R_2) + w * K_{PET}(R_1, R_2).$$

where K_{HF} is a feature based kernel that uses a heterogeneous set of features, K_{SL} is the Shallow Linguistic (SL) kernel proposed by Giuliano et al. [16], and K_{PET} stands for the Path-enclosed Tree (PET) kernel [28]. *w* is a multiplicative constant that allows the hybrid kernel to assign more (or less) weight to the information obtained using tree structures depending on the corpus. We exploit the SVM-Light-TK toolkit [29,19] for kernel computation. The parameters are tuned by doing 5-fold cross validation on the training data.

5.4. DDI type classification

The second step is to classify the extracted DDIs into different categories. We train 4 separate models for each of the DDI types (one vs all) to predict the class label of the extracted DDIs. During this training, all the negative instances from the training data are removed. The filtering techniques described in Sections 5.1 and 5.2 are not used in this stage.

Once the above models are trained, they are applied on the DDIs extracted from the test data. The class label of the model which has the highest confidence score for an extracted DDI instance is assigned to such instance.

6. Experiments on the DDI-Extraction-2011 corpus

6.1. Dataset

The DDI-Extraction-2011 challenge[20] required the automatic identification of DDI from biomedical articles. Only the intra-sentential DDI (i.e. DDI within single sentence boundaries) are considered. The challenge corpus [35] is divided into training and evaluation dataset. Initially released training data consist of 435 abstracts and 4267 sentences, and were annotated with 2402 DDI. During the evaluation phase, a dataset containing 144 abstracts and 1539 sentences was provided to the participants as the evaluation data. Both datasets contain drug annotations, but only the training dataset has DDI annotations.

These datasets are made available in two formats: the so-called *unified* format and the *MMTx* format. The unified format contains only the tokenized sentences, while the MMTx format contains the tokenized sentences along with POS tag for each token.

We used the unified format data. In both training and evaluation datasets, there are some missing special symbols, perhaps due to encoding problems. The position of these symbols can be identified by the presence of the question mark "*?*" symbol. For example:

> ⟨*sentence id = "DrugDDI.d554.s14" origId = "s14" text = "Ergotamine or dihydroergotamine***?***acute ergot toxicity characterized by severe peripheral vasospasm and dysesthesia.*"⟩

6.2. Results using our hybrid approach FBM–KBM

We split the original development corpus into two parts: the first part contains 63% of the documents (i.e. 276 docs) containing around 67% of the "true" DDI pairs (i.e. 1603) and was used for tuning the systems. The remaining documents belong to the second part which was used as a test corpus for performance evaluation. Both systems of the hybrid approach were trained and evaluated using these splits of the development corpus (cf. Table 1 shows their respective results).

The recall of the union (on the positive DDI) of the outputs of each approach was higher than the individual output of the systems. We also calculated results for the intersection (only common positive DDI) of the outputs which decreased the performance. It is also important to note that the feature-based method (FBM) reached higher precision while the kernel-based method (KBM) obtained higher recall.

Table 2 shows the evaluation results for the proposed approaches on the final challenge evaluation corpus. The union of outputs of the systems has produced an F_1 score of **0.6398** which is better than the individual results. The behavior of precision and recall obtained by the two approaches is the same as observed on the development corpus (better precision for the feature-based approach and better recall for the kernel-based approach), however, the F_1 score of the kernel-based approach is quite close (F_1 score of *0.6365*) to that of the union.

6.3. Results using K_{Hybrid} (i.e. the 1st step of the two-step approach)

We used the K_{Hybrid} (which is used in the 1st step of our two-step approach for DDI extraction) to do similar experiments on the DDI-Extraction-2011 corpus. The results (see Table 3) show that the more advanced linguistically informed K_{Hybrid} based RE approach outperforms the hybrid approach which is the union of KBM and FBM described before. In fact, the results of K_{Hybrid} are also significantly better than the results of the best system in the DDI-Extraction-2011 shared task.

7. Experiments on the DDI-Extraction-2013 corpus

7.1. Dataset

The DDI-Extraction-2013 shared task includes two tasks:

- Task 9.1: Recognition and classification of drug names.
- Task 9.2: Extraction and Classification of drug–drug interactions.

[20] http://labda.inf.uc3m.es/DDIExtraction2011/.

Table 1
DDI-Extraction-2011. Experimental results when trained on 63% of the original training documents and tested on the remaining (FBM: Feature-Based Method, KBM: Kernel-Based Method).

	FBM	KBM	Union	Intersection
Precision	0.5910	0.4342	0.4218	0.6346
Recall	0.3640	0.5277	0.6083	0.2821
F_1 score	0.4505	0.4764	**0.4982**	0.3906

Bold value correspond to the best results.

Table 2
DDI-Extraction-2011. Evaluation results provided by the challenge organizers (FBM: Feature-Based Method, KBM: Kernel-Based Method).

	FBM	KBM	Union
True positive	319	513	532
False positive	133	344	376
False negative	436	242	223
True negative	6138	5927	5895
Precision	**0.7058**	0.5986	0.5859
Recall	0.4225	0.6795	**0.7046**
F_1 score	0.5286	0.6365	**0.6398**

Bold values correspond to the best results.

The DDI-Extraction-2013 corpus contains 1017 medical texts: (i) 233 MedLine abstracts on drug–drug interactions and (ii) 784 documents describing drug–drug interactions from the DrugBank database, in order to deal with different types of texts and language styles. The corpus is annotated with drug–drug interactions and pharmacological substances.

Participants were asked to not only extract DDI but also classify them into one of four pre-defined classes: `advise`, `effect`, `mechanism` and `int`. A detailed description of the task settings and data can be found in [33]. Evaluation results are reported using the standard Precision, Recall and *F*1 score metrics.

7.2. Task 9.1: Recognition and classification of drug names

The first task[21] focuses on the extraction and classification of four types of pharmacological entities:

- **Drug** is any chemical entity that is used for treatment, diagnosis of disease, prevention or cure and is approved for human use.
- **Brand** is any drug that was developed firstly by a pharmaceutical company.
- **Group** is any term that describes a chemical or pharmacological relationship of drugs.
- **Drug_n** is any active substance that has not been approved for human use.

The organizers of the task proposed several matching criteria for the evaluation of the results, which are:

- **Strict evaluation**: correct mentions must have the correct start offset and end offset as well as the correct entity type.
- **Type matching**: correct mentions can overlap the reference mentions if they have the correct entity type.
- **Exact boundary matching**: correct mentions must have the same start and end offsets of the reference mentions, regardless of the entity type (used to evaluate strict boundary detection).

Table 3
DDI-Extraction-2011. Comparison of results on the official test set using the hybrid approach and the K_{Hybrid} kernel based RE approach.

	P	*R*	*F*1 score
Hybrid approach (i.e. union of KBM and FBM)	0.5859	0.7046	0.6398
Proposed K_{Hybrid}	**0.6001**	**0.7432**	**0.6640**

Bold values correspond to the best results.

- **Partial boundary matching**: correct mentions must have an overlap with the reference mentions, regardless of the entity type (used to evaluate partial boundary detection).

We use different combinations of the features and two different tools for annotating the words with their part-of-speech tags (POS-tags) and lemma.

- **Run 1**: All the linguistic and token features (see Section 3.2) as well as the features for annotating the words as stop words, abbreviations and medical units of measurement are used. **TreeTagger**[22] is used for annotating the words with part-of-speech and lemma information.
- **Run 2**: Uses the features of the first run and the **StanfordTagger**[23] tool for estimating the boundaries of words, their lemma and their part-of-speech tags instead of TreeTagger (used in the first run).
- **Run 3**: Uses the features of the second run with a feature for annotating the words in the BIO format of the **Drug** class, using lists of drug names.
- **Run 4**: Uses all the features of the third run and a feature for annotating the words in the BIO format of the **Ingredient** class using a list of drugs' ingredients.

Table 4 presents the results according to the four criteria. The *F*1 score obtained for run 1 is 0.69 for partial matching and this score slightly decreased to 0.67 for exact matching in the Drugbank and Medline datasets. This means that most of the tagged mentions of our system have the correct start and end offset. *F*1 score decreases when the system checks the entity type (0.64 *F*1 for type matching) and for exact boundaries (0.58 *F*1 for strict matching).

In Run 2, we use the StanfordTagger instead of TreeTagger for finding the mentions boundaries as well as their part-of-speech tags and their lemmas. The *F*1 scores corresponding to exact and partial boundary matching slightly increased by 0.01 point overall. For the Drugbank dataset the *F*1 score for the exact and partial boundaries matching increased respectively by +0.3 and +0.4. These results suggest that StanfordTagger identifies the boundaries of the mentions more effectively than TreeTagger. The final F1 scores for strict matching and type matching has therefore also increased by +0.02 points overall and by +0.05 points on the Drugbank dataset.

In Run 3, the *F*1 scores for strict and type matching are slightly better than the scores of Run 2 with a +0.02 increase on both benchmarks. This may be explained by the addition of lists of drug names for annotation of the words with the BIO tags for the ***Drug*** class. The *F*1 scores for the partial and exact boundaries matching remained stable as expected.

In Run 4, the *F*1 score of strict and type matching are higher than those of Run 3. More precisely, *F*1 scores for strict evaluation slightly increased from 0.62 to 0.63 and the *F*1 score for type matching increased from 0.68 to 0.69. As the names of the ingredients can be related to the names of drugs, the CRF algorithm had an additional clue to predict more effectively whether a word belongs to a drug name or not.

[21] http://www.cs.york.ac.uk/semeval-2013/task9/data/uploads/task-9.1-drug-ner.pdf.

[22] http://www.cis.uni-muenchen.de/~schmid/tools/TreeTagger.

[23] http://nlp.stanford.edu/software/tagger.shtml.

Table 4
DDI-Extraction-2013 Task 9.1. Results according to different criteria.

Criteria	Run	DrugBank and MedLine			MedLine			DrugBank		
		P	*R*	*F*	*P*	*R*	*F*	*P*	*R*	*F*
Strict evaluation	1	0.78	0.46	0.58	0.7	0.23	0.35	0.9	0.59	0.71
	2	0.79	0.48	0.6	0.74	**0.25**	**0.37**	0.91	0.64	0.76
	3	0.83	0.49	0.62	0.73	0.24	0.36	0.93	0.65	0.77
	4	0.85	**0.5**	**0.63**	0.73	0.24	0.36	0.93	**0.7**	**0.8**
Type matching	1	0.86	0.51	0.64	0.82	0.27	0.41	0.96	0.63	0.76
	2	0.88	0.53	0.66	0.86	**0.29**	**0.44**	0.97	0.69	0.81
	3	0.9	**0.54**	0.68	0.86	0.28	0.43	0.99	0.69	0.81
	4	0.93	**0.54**	**0.69**	0.86	0.28	0.42	0.98	**0.74**	**0.84**
Exact boundary matching	1	0.9	0.53	0.67	0.87	**0.29**	0.43	0.93	0.61	0.73
	2	0.89	**0.54**	**0.68**	0.86	**0.29**	**0.44**	0.91	0.64	0.76
	3	0.91	**0.54**	**0.68**	0.86	0.28	0.42	0.94	0.66	0.78
	4	0.91	0.53	0.67	0.85	0.27	0.42	0.95	**0.71**	**0.81**
Partial boundary matching	1	0.9	0.56	0.69	0.87	0.31	**0.46**	0.93	0.63	0.75
	2	0.89	**0.57**	**0.7**	0.86	**0.32**	**0.46**	0.93	0.68	0.79
	3	0.91	**0.57**	**0.7**	0.86	0.3	0.45	0.94	0.68	0.79
	4	0.91	0.56	0.69	0.85	0.3	0.44	0.95	**0.73**	**0.83**

Bold values correspond to the best results.

Table 5 illustrates the results of precision, recall and *F*1 scores for each entity type (Drug, Brand, Group and Drug_n). According to these results, our system for drug name recognition predicted correctly entities of type "Brand" with 0.92 *F*1 score and the entity of type "Drug" with 0.72 *F*1 score. Our system is less effective for predicting the Group entities (0.63 *F*1 score) than the Brand or Drug types. It also failed to identify the words that are related to Drug_n entities. This last aspect may be caused by the absence of relevant features for chemical compounds (e.g. features based on the ChEBI[24] dictionary, which will be added in coming work).

Table 6 represents the macro-average *F*1 score computed over all entity types. It shows that Run 3 is the most effective configuration for drug name recognition among the 4 runs overall and for the DrugBank dataset. Run 2 is the most effective run for identifying pharmaceutical substances in the MedLine dataset. It is also observed that the highest score for all metrics is obtained on the DrugBank dataset. Actually, the MedLine dataset contains only abstracts and has therefore fewer medical texts than the DrugBank dataset which usually contain more details and cues to recognize drug names.

Overall, our method based on Conditional Random Fields (CRFs) combined with linguistic and semantic features reached a macro-average *F*1 score of 0,57. Grego et al. [17] achieved almost the same score (0.577 *F*1) as ours. They used CRFs combined with linguistic features and database annotations. They also used a resolution method that takes as input the annotated named entities and provides the most relevant ChEBI terms from the ChEBI ontology. Rocktäschel et al. [31] achieved the best *F*1 score (0.652) in the DDI-Extraction-2013 task. They constructed a named entity recognition system using CRFs combined with linguistic features and semantic features derived from Jochem, PHARE and ChEBI ontologies.

7.3. Task 9.2: Extraction and classification of drug–drug interactions

The second task[25] focuses on the extraction and classification of DDI. As reported by the organizers in the task description paper [33], the results obtained by our system for both DDI extraction and classification are significantly higher than all the other participants in the shared task.

The task 9.2 data include two types of texts: texts taken from the DrugBank database and texts taken from MedLine abstracts. During training we used both types of text together.

The Charniak–Johnson reranking parser [4], along with a self-trained biomedical parsing model [27], has been used for tokenization, POS-tagging and parsing of the sentences. Then the parse trees are processed by the Stanford parser [21] to obtain syntactic dependencies. The Stanford parser often skips some syntactic dependencies in output. We use the rules proposed in [9] to recover some of such dependencies. We use the same techniques for unknown characters (if any) as described in [8].

Our system uses the SVM-Light-TK toolkit[26] [29,19] for computation of the hybrid kernels. The ratio of negative and positive examples has been used as the value of the cost-ratio-factor parameter. The SL kernel is computed using the jSRE tool.[27]

The K_{HF} kernel can exploit non-target entities to extract important clues [11]. So, we use a publicly available state-of-the-art NER system called BioEnEx [7] to automatically annotate both the training and the test data with disease mentions.

Table 7 shows the results of 5-fold cross validation for DDI detection on the training data. As we can see, the usage of the LIS and LII filtering techniques improves both precision and recall.

We submitted three runs for the DDI-Extraction-2013 shared task. The only difference between the three runs concerns the default class label (i.e. the class chosen when none of the separate models assigns a class label to a predicted DDI). Such default class label is `int`, `effect` and `mechanism` for run 1, 2 and 3 respectively. According to the official results provided by the task organizers, our best result was obtained by run 2 (shown in Table 8).

According to the official results, the performance for `advise` is very low (F_1 0.29) in MedLine texts, while the performance for `int` is comparatively much higher (F_1 0.57) with respect to the one of the other DDI types. In comparison, the performance for `int` is much lower (F_1 0.55) in DrugBank texts with respect to the one of the other DDI types.

In MedLine test data, the number of `effect` (62) and `mechanism` (24) DDIs is much higher than that of `advise` (7) and `int` (2). On the other hand, in DrugBank test data, the different DDIs are more evenly distributed – `effect` (298), `mechanism` (278), `advise` (214) and `int` (94).

Initially, it was not clear to us why our system (as well as other participants) achieves so much higher results on the DrugBank sentences in comparison to MedLine sentences. Statistics of the average number of words show that the length of the two types

[24] http://www.ebi.ac.uk/chebi/downloadsForward.do.
[25] http://www.cs.york.ac.uk/semeval-2013/task9/data/uploads/task-9.2-ddi-extraction.pdf.
[26] http://disi.unitn.it/moschitti/Tree-Kernel.htm.
[27] http://hlt.fbk.eu/en/technology/jSRE.

Table 5
DDI-Extraction-2013 Task 9.1. Results according to each entity type (Drug, Brand, Groups and Drug_n).

Entity type	Run	DrugBank and MedLine			MedLine			DrugBank		
		P	*R*	*F*	*P*	*R*	*F*	*P*	*R*	*F*
Drug	1	0.86	0.58	0.69	0.8	0.43	0.56	0.92	0.62	0.74
	2	0.85	0.6	0.7	0.83	**0.48**	**0.61**	0.91	0.65	0.76
	3	0.87	0.6	0.71	0.79	0.46	0.58	0.94	0.65	0.77
	4	0.9	**0.61**	**0.72**	0.81	0.45	0.58	0.94	**0.72**	**0.82**
Brand	1	0.98	0.68	0.8	0	0	0	0.97	0.64	0.77
	2	1	0.76	0.87	0	0	0	1	0.79	0.88
	3	1	0.85	0.92	0	0	0	1	**0.87**	**0.93**
	4	0.98	**0.86**	**0.92**	0	0	0	1	**0.87**	**0.93**
Group	1	0.83	0.45	0.59	0.87	0.14	0.25	0.85	0.51	0.63
	2	0.85	0.48	0.62	0.76	**0.14**	**0.24**	0.88	**0.57**	**0.69**
	3	0.86	**0.5**	**0.63**	0.75	0.13	0.23	0.88	0.55	0.68
	4	0.86	**0.5**	**0.63**	0.75	0.13	0.23	0.88	**0.57**	**0.69**
Drug_n	1	0	0	0	1	0.01	0.02	0	0	0
	2	0	0	0	1	0.01	0.02	0	0	0
	3	0	0	0	1	0.01	0.02	0	0	0
	4	0	0	0	1	0.01	0.02	0	0	0

Bold values correspond to the best results.

Table 6
DDI-Extraction-2013 Task 9.1. Macro-average measures for each run.

Run	DrugBank and MedLine			MedLine			DrugBank		
	P	*R*	*F*	*P*	*R*	*F*	*P*	*R*	*F*
1	0.67	0.43	0.52	0.67	0.15	0.21	0.68	0.44	0.54
2	0.68	0.46	0.55	0.65	**0.16**	**0.22**	0.7	0.5	0.58
3	0.68	0.49	0.56	0.63	0.15	0.21	0.7	0.52	0.59
4	0.68	**0.49**	**0.57**	0.64	0.15	0.21	0.71	**0.54**	**0.61**

Bold values correspond to the best results.

of training sentences are substantially similar (DrugBank: 21.2, MedLine: 22.3). It is true that the number of the training sentences for the former is almost 5.3 times higher than the latter. But it could not be the main reason for such high discrepancies.

So, we turned our attention to the presence of the cue words. In the 4,683 sentences of the DrugBank training set (which have at least one drug mention), we found that the words "increase" and "decrease" are present in 721 and 319 sentences respectively. On the other hand, in the 877 sentences of the MedLine training set (which have at least one drug mention), we found that the same words are present in only 67 and 40 sentences respectively. In other words, the presence of these two important cue words in the DrugBank sentences is twice more likely than that in the MedLine sentences. We assume similar observations might be also possible for other cue words. Hence, this is probably the main reason why the results are so much better on the DrugBank sentences.

8. Discussion and future work

Our approach to drug name recognition obtained encouraging results with respect to systems participating in the shared task of DDI-Extraction-2013, especially considering that we made the choice to use only publicly available lists as semantic features, without relying on unavailable (*ad hoc*) tools or ontologies. In future work we are considering to test the impact of annotations from open-domain ontologies like DBpedia or YAGO when used as features in the drug name recognition process.

Table 7
DDI-Extraction-2013 Task 9.2. Comparison of results for DDI detection on the training data using 5-fold cross validation. Parameter tuning is not done during these experiments.

	P	*R*	F_1
K_{Hybrid}	0.66	0.80	0.72
LIS filtering + K_{Hybrid}	0.67	0.80	0.73
LIS filtering + LII filtering + K_{Hybrid}	**0.68**	**0.82**	**0.74**

Bold values correspond to the best results.

Table 8
DDI-Extraction-2013 Task 9.2. Official results of the best run of our system (run 2).

	P	*R*	F_1
All text			
DDI detection only	0.79	0.81	0.80
Detection and classification	0.65	0.66	0.65
DrugBank text			
DDI detection only	0.82	0.84	0.83
Detection and classification	0.67	0.69	0.68
MedLine text			
DDI detection only	0.56	0.51	0.53
Detection and classification	0.42	0.38	0.40

For DDI extraction we studied the combination of Feature-Based Methods (FBM) and Kernel-Based Methods (KBM). Our hybrid system was ranked second in the DDI-Extraction-2011 task. The results show that the KBM outperformed the FBM. We also observed that the union of results of both methods led only to slight improvement of 0.0033 w.r.t. KBM according to *F*1 score.

The results obtained by the combination of FBM and KBM and the fact that the precision of the FBM was 11% higher than KBM (cf. Table 2) suggest that the DDIs found only by the FBM are mostly incorrect, i.e. that the KBM already found most of the correct DDIs that were retrieved by the FBM. Therefore, the logical track of enhancement would rather be to use a method that helps identifying the false positives retrieved by the KBM.

We significantly improved our technique for DDI extraction by developing a high performance hybrid kernel, K_{Hybrid}, and extended extraction to DDI classification (i.e. our two-step approach). This state-of-the-art approach outperformed the results obtained by all the other participating teams in the DDI-Extraction-2013 shared task by a wide margin.

In future work, we will study the performance of these different methods for the detection of Adverse Drug Reactions (ADR), which is one of the priority areas considered by the World Health Organization (2002).[28] We will particularly address two main challenges to text mining for pharmacovigilance:

[28] http://apps.who.int/iris/bitstream/10665/42493/1/a75646.pdf?ua=1.

- Data interpretation and integration. Using different sources of information about pharmacovigilance that contain a huge amount of data and provide complementary knowledge (e.g. scientific papers, forums, social networks, linked data).
- Personalization according to the patient profile. Reactions of patients to drugs can be very different. We think that it is crucial to include patient characteristics in the analysis process, whether by model-based inference or feature-based learning.

9. Conclusion

Text mining can be an efficient and scalable solution for the analysis of medical corpora such as scientific articles or clinical records in order to understand and support pharmacovigilance. This paper reported our experiments on text mining techniques for the extraction and classification of drugs and drug–drug interactions.

We presented our hybrid method for DDI extraction that combines two different machine learning methods to extract DDI: (i) a feature-based method that uses a SVM classifier with a set of lexical, morphosyntactic and semantic features and (ii) a kernel-based method that uses a kernel which is a composition of a *mildly extended dependency tree* kernel, a *phrase structure tree* kernel, and a *shallow linguistic* kernel. We participated in the *DDI-Extraction-2011 challenge* and we obtained 0.6398 *F*1 score, the second best results in the shared task.

For drug name recognition, we presented our feature-based method that identifies and classifies drugs into four classes. We evaluated our method on the Drug recognition corpus of the *DDI-Extraction challenge at SemEval 2013* and obtained encouraging results: 0.57 *F*1 score on all texts and 0.61 *F*1 score on the DrugBank corpus.

Finally, we presented our two-step approach for the detection and classification of DDI. Our approach outperformed all the other participating teams in the *DDI Detection and Classification task at SemEval 2013* with 0.80 *F*1 score for detection only and 0.658 *F*1 score for detection and classification. The central component of the proposed approach is a state-of-the-art hybrid kernel. Our approach also exploits the scope of negation cues and the semantic roles of the involved entities.

The three systems described in this paper and the three available corpora used to train and evaluate our systems represent a first step towards supporting pharmacovigilance through text mining. Nevertheless, more information should be taken into account such as patient characteristics (e.g. age, history of diseases and drugs) and more document sources should be explored. In future work, we plan to exploit data integration techniques in order to use different sources of information (e.g. articles, websites, linked data). We also plan to develop NLP techniques adapted to each information source for the recognition of drugs, side effects, information about patients and for the extraction of DDI and other relations between drugs and side effects. Future work will include also the design and development of a web service that incorporates our three systems, with online annotation and visualization components.

Conflict of interest

None declared.

References

[1] A. Ben Abacha, P. Zweigenbaum, A hybrid approach for the extraction of semantic relations from MEDLINE abstracts, in: 12th International Conference on Computational Linguistics and Intelligent Text Processing, CICLing 2011, Tokyo, Japan, 2011, pp. 139–150, http://dx.doi.org/10.1007/978-3-642-19400-9, doi:http://dx.doi.org/10.1007/978-3-642-19400-9.

[2] A. Ben Abacha, P. Zweigenbaum, Medical entity recognition: a comparison of semantic and statistical methods, in: BioNLP 2011 Workshop, Association for Computational Linguistics, Portland, Oregon, USA, 2011, pp. 56–64 <http://www.aclweb.org/anthology/W11-0207>.

[3] C.C. Chang, C.J. Lin, LIBSVM: A Library for Support Vector Machines, 2001 <http://www.csie.ntu.edu.tw/∼cjlin/libsvm>.

[4] E. Charniak, M. Johnson, Coarse-to-fine n-best parsing and MaxEnt discriminative reranking, in: Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL 2005), 2005.

[5] Y. Chen, F. Liu, B. Manderick, Extract protein–protein interactions from the literature using support vector machines with feature selection, Biomed. Eng., Trends, Res. Technol. (2011).

[6] F.M. Chowdhury, A. Ben Abacha, A. Lavelli, P. Zweigenbaum, Two different machine learning techniques for drug–drug interaction extraction, in: Isabel Segura-Bedmar, Paloma Martinez, Daniel Sanchez-Cisneros (Eds.), Proceedings DDIExtraction2011, First Challenge Task on Drug–Drug Interaction Extraction 2011 (SEPLN 2011 Satellite Workshop), CEUR Workshop Proceedings, Association for Computational Linguistics, Huelva, Spain, volume 761, 2011, pp. 19–26.

[7] M. Chowdhury, A. Lavelli, Disease mention recognition with specific features, in: Proceedings of the 2010 Workshop on Biomedical Natural Language Processing, Uppsala, Sweden, 2010, pp. 83–90 <http://www.aclweb.org/anthology/W10-1911>.

[8] M. Chowdhury, A. Lavelli, Drug–drug interaction extraction using composite kernels, in: Proceedings of the 1st Challenge task on Drug–Drug Interaction Extraction (DDIExtraction 2011), Huelva, Spain, 2011, pp. 27–33.

[9] M. Chowdhury, A. Lavelli, Combining tree structures, flat features and patterns for biomedical relation extraction, in: Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics (EACL 2012), Avignon, France, 2012, pp. 420–429.

[10] M. Chowdhury, A. Lavelli, Impact of less skewed distributions on efficiency and effectiveness of biomedical relation extraction, in: Proceedings of the 24th International Conference on Computational Linguistics (COLING 2012), Mumbai, India, 2012.

[11] M. Chowdhury, A. Lavelli, Exploiting the scope of negations and heterogeneous features for relation extraction: a case study for drug–drug interaction extraction, in: Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technology (NAACL 2013), Atlanta, USA, 2013.

[12] M.F.M. Chowdhury, A. Lavelli, FBK-irst: a multi-phase kernel based approach for drug–drug interaction detection and classification that exploits linguistic information, in: Second Joint Conference on Lexical and Computational Semantics (∗SEM), Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013), vol. 2, Association for Computational Linguistics, Atlanta, Georgia, USA, 2013, pp. 351–355 <http://www.aclweb.org/anthology/S13-2057>.

[13] M.F.M. Chowdhury, A. Lavelli, A. Moschitti, A study on dependency tree kernels for automatic extraction of protein–protein interaction, in: Proceedings of BioNLP 2011 Workshop, Association for Computational Linguistics, Portland, Oregen, USA, 2011, pp. 124–133.

[14] A. Coulet, Y. Garten, M. Dumontier, R.B. Altman, M.A. Musen, N.H. Shah, Integration and publication of heterogeneous text-mined relationships on the semantic web, J. Biomed. Seman. 2 (2011) S10. <http://www.jbiomedsem.com/content/2/S2/S10>.

[15] A. Culotta, J. Sorensen, Dependency tree kernels for relation extraction, in: Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics (ACL '04), Barcelona, Spain, 2004.

[16] C. Giuliano, A. Lavelli, L. Romano, Exploiting shallow linguistic information for relation extraction from biomedical literature, in: Proceedings of the 11th Conference of the European Chapter of the Association for Computational Linguistics (EACL'2006), Trento, Italy, 2006, pp. 401–408.

[17] T. Grego, F. Pinto, F.M. Couto, LASIGE: Using Conditional Random Fields and ChEBI Ontology, Association for Computational Linguistics, 2013, pp. 660–666 <http://aclweb.org/anthology/S13-2109>.

[18] K.M. Hettne, R.H. Stierum, M.J. Schuemie, P.J.M. Hendriksen, B.J.A. Schijvenaars, E.M. van Mulligen, J. Kleinjans, J.A. Kors, A dictionary to identify small molecules and drugs in free text, Bioinformatics 25 (2009) 2983–2991. http://dx.doi.org/10.1093/bioinformatics/btp535.

[19] T. Joachims, Making large-scale support vector machine learning practical, in: Advances in Kernel Methods: Support Vector Learning, MIT Press, Cambridge, MA, USA, 1999, pp. 169–184.

[20] K. Johnell, I. Klarin, The relationship between number of drugs and potential drug–drug interactions in the elderly: a study of over 600,000 elderly patients from the swedish prescribed drug register, Drug Saf. 30 (2007) 911–918.

[21] D. Klein, C.D. Manning, Accurate unlexicalized parsing, in: Proceedings of the 41st Annual Meeting on Association for Computational Linguistics (ACL '03), Association for Computational Linguistics, Sapporo, Japan, 2003, pp. 423–430.

[22] J.D. Lafferty, A. McCallum, F.C.N. Pereira, Conditional random fields: probabilistic models for segmenting and labeling sequence data, in: Proceedings of the Eighteenth International Conference on Machine Learning (ICML 2001), June 28–July 1, 2001, Williams College, Williamstown, MA, USA, 2001, pp. 282–289.

[23] E. Landau, Jackson's Death Raises Questions About Drug Interactions [Published in CNN; June 26, 2009], 2009 <http://edition.cnn.com/2009/HEALTH/06/26/jackson.drug.interaction.caution/index.html>.
[24] L. Li, J. Ping, D. Huang, Protein–protein interaction extraction from biomedical literatures based on a combined kernel, J. Inf. Comput. Sci. 7 (5) (2010) 1065–1073.
[25] P. de Matos, R. Alcántara, A. Dekker, M. Ennis, J. Hastings, K. Haug, I. Spiteri, S. Turner, C. Steinbeck, Chemical entities of biological interest: an update, Nucl. Acids Res. 38 (2010) 249–254. http://dx.doi.org/10.1093/nar/gkp886.
[26] A. McCallum, W. Li, Early results for named entity recognition with conditional random fields, feature induction and web-enhanced lexicons, in: Proceedings of the Seventh Conference on Natural Language Learning, CoNLL 2003, Held in Cooperation with HLT-NAACL 2003, May 31–June 1, 2003, Edmonton, Canada, 2003, pp. 188–191.
[27] D. McClosky, Any Domain Parsing: Automatic Domain Adaptation for Natural Language Parsing. Ph.D. thesis. Department of Computer Science, Brown University, 2010.
[28] A. Moschitti, A study on convolution kernels for shallow semantic parsing, in: Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics (ACL '04), Barcelona, Spain, 2004.
[29] A. Moschitti, Efficient convolution kernels for dependency and constituent syntactic trees, in: J. Fürnkranz, T. Scheffer, M. Spiliopoulou (Eds.), Machine Learning: ECML 2006, Lecture Notes in Computer Science, vol. 4212, Springer, Berlin/Heidelberg, 2006, pp. 318–329.
[30] J. Payne, A Dangerous Mix [Published in The Washington Post; February 27, 2007], 2007 <http://www.washingtonpost.com/wp-dyn/content/article/2007/02/23/AR2007022301780.html>.
[31] T. Rocktäschel, T. Huber, M. Weidlich, U. Leser, WBI-NER: the impact of domain-specific features on the performance of identifying and classifying mentions of drugs, in: Proceedings of the 7th International Workshop on Semantic Evaluation (SemEval 2013), Association for Computational Linguistics, Sapporo, Japan, 2013, pp. 356–363.
[32] T. Rocktäschel, M. Weidlich, U. Leser, ChemSpot: a hybrid system for chemical named entity recognition, Bioinformatics 28 (2012) 1633–1640. http://dx.doi.org/10.1093/bioinformatics/bts183.
[33] I. Segura-Bedmar, P. Martínez, M. Herrero Zazo, SemEval-2013 task 9: extraction of drug–drug interactions from biomedical texts (DDIExtraction 2013), in: Second Joint Conference on Lexical and Computational Semantics (∗SEM), Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013), vol. 2, Association for Computational Linguistics, Atlanta, Georgia, USA, 2013. pp. 341–350 <http://www.aclweb.org/anthology/S13-2056>.
[34] I. Segura-Bedmar, P. Martínez, C.d. Pablo-Sánchez, Extracting drug–drug interactions from biomedical texts, BMC Bioinf. 11 (suppl. 5) (2010) 9.
[35] I. Segura-Bedmar, P. Martínez, C.D. Pablo-Sánchez, Using a shallow linguistic kernel for drug–drug interaction extraction, J. Biomed. Inf. 44 (5) (2011) 789–804.
[36] I. Segura-Bedmar, P. Martínez, M. Herrero-Zazo, Lessons learnt from the DDIExtraction-2013 shared task, J. Biomed. Inf. 51 (2014) 152–164. <http://www.sciencedirect.com/science/article/pii/S1532046414001245>.
[37] M. Song, H. Yu, W. Han, Combining active learning and semi-supervised learning techniques to extract protein interaction sentences, in: International Workshop on Data Mining in Bioinformatics, 2010.